AD-A069 756 AIR FORCE ENVIRONMENTAL TECHNICAL APPLICATIONS CENTER--ETC F/G 9/2
USAFETAC ANALYST/PROGRAMMER HANDBOOK; IBM 360 DOS USER'S GUIDE.(U)
MAR 79 M L FREIMUND

UNCLASSIFIED USAFETAC/TN-79/001 NL

| OF |
AD
A069756

END
DATE
FILMED
7-79
DDC

USAFETAC/TN-79/001

LEVEL II

USAFETAC ANALYST/PROGRAMMER HANDBOOK

IBM 360 DOS User's Guide

Marvin L. Freimund, TSgt, USAF

March 1979

Approved for public release; distribution unlimited.

UNITED STATES AIR FORCE
AIR WEATHER SERVICE (MAC)

USAF ENVIRONMENTAL
TECHNICAL APPLICATIONS CENTER

SCOTT AIR FORCE BASE, ILLINOIS 62225

REVIEW AND APPROVAL STATEMENT

This report approved for public release. There is no objection to unlimited distribution of this report to the public at large, or by DDC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

MARVIN L. FREIMUND, TSgt, USAF
Systems Analyst
Author

MICHAEL J. MC GLINN, Capt, USAF
Chief, Integration, Control, and
Standardization Section
Reviewing Officer

FOR THE COMMANDER

WALTER S. BURGMANN
Scientific and Technical
Information Officer (STINFO)

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

(14) **REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: USAFETAC/TN-79/001
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) USAFETAC ANALYST/PROGRAMMER HANDBOOK; IBM 360 DOS User's Guide
5. TYPE OF REPORT & PERIOD COVERED: (9) Technical rept.
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) Marvin L. Freimund, TSgt, USAF
8. CONTRACT OR GRANT NUMBER(s): (12) 21p.

9. PERFORMING ORGANIZATION NAME AND ADDRESS
U.S. Air Force Environmental Technical Applications Center
Scott AFB, Illinois 62225

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
U.S. Air Force Environmental Technical Applications Center
Scott AFB, Illinois 62225

12. REPORT DATE: (11) Mar[illegible]79
13. NUMBER OF PAGES: 22

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*: Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*
Computers
Computer system components disk operating system
Disk operating system
IBM 360

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This handbook provides a brief description of the IBM 360/44 computer system at USAFETAC Scott AFB, Illinois. Descriptions of the system configuration, system organization, and system programming are included.

400 945

DD FORM 1 JAN 73 1473

PREFACE

The IBM 360/44 is a complex and versatile computer; however, it has some definite restrictions and limitations. It is possible to program and use the system without knowing a great deal about it, just as it is possible to operate an automobile with little actual knowledge of the machine. If one desires to use either a computer or an automobile efficiently and effectively and is interested in staying out of trouble, some knowledge of the inner workings of the system is indespensible.

This handbook is designed to give the user of the US Air Force Environmental Technical Applications Center (USAFETAC) computer facilities the knowledge necessary to get the most out of the IBM 360/44. It presupposes a basic knowledge of computer science and systems architecture such as that normally found in a first course in programming or quantitative methods. It is a reference manual, not a text. It is not intended for the novice user with very little or no computer background. The average USAFETAC user, however, should find it a valuable tool to assist in making the IBM 360/44 a friend rather than an adversary.

TABLE OF CONTENTS

		Page
Chapter 1	SYSTEM CONFIGURATION	1
1.1.	General	1
1.2.	Processing Unit	1
1.3.	TELEX 5312 Direct Access Storage Devices	1
1.4.	IBM 1052 Console Printer/Keyboard	1
1.5.	Magnetic-Tape Units	2
1.5.1.	STC 3470 Magnetic-Tape Unit (9-Track)	2
1.5.2.	IBM 2401 Magnetic-Tape Unit (7-Track)	2
1.5.3.	Tape Assignments	2
1.6.	IBM 2540 Card Read/Punch	2
1.7.	IBM 1403 Model N1 Printer	3
Chapter 2	SYSTEM ORGANIZATION	4
2.1.	General	4
2.2.	Supervisor	4
2.3.	Job Control	4
2.3.1.	Preparing Programs for Execution	5
2.3.2.	Symbolic I/O Assignment	5
2.3.3.	Job Control Statements	5
2.4.	GRASP	6
2.4.1.	EPROC	6
2.4.2.	GRASPCTL	7
2.4.3.	LINEUP	7
2.5.	Disk Organization	8
Chapter 3	SYSTEM/360 PROGRAMMING	10
3.1.	General	10
3.2.	Language Translators	10
3.2.1.	Assembler	12
3.2.2.	COBOL	12
3.2.3.	FORTRAN	13
3.3.	Linkage Editor	13
3.3.1.	Catalog Programs in Core Image Library	14
3.3.2.	Load-and-Execute	14
3.3.3.	Assemble- or Compile-and-Execute	14
3.4.	Program Library	14
APPENDIX	LISTING OF APPLICABLE IBM SYSTEM REFERENCE MANUALS	15
LIST OF ABBREVIATIONS AND ACRONYMS		15

LIST OF ILLUSTRATIONS

Figure 2-1.	USAFETAC Disk Pack Assignments	9
Figure 3-1.	USAFETAC System/360 DOS Facilities	11

Chapter 1

SYSTEM CONFIGURATION

1.1. General. Model 44 of the IBM System/360 is tailored to handle all relatively small to medium sized scientific applications, advanced data acquisition, and process control applications. Although the Model 44 is thus specialized, its inclusion within the context of System/360 philosophy provides for a very large addressing capability, a wide range of high-speed storage capacities, and Input/Output (I/O) multiplexing on channels of low and high data-transfer rates.

The following sections briefly describe the IBM System/360 and peripheral configuration at USAFETAC. For a more detailed description of any of the system components refer to the applicable IBM publications listed in the Appendix.

1.2. Processing Unit. As in all other models of System/360, the smallest addressable data unit is the byte, consisting of 8 bits. A 9th bit, called the parity bit (not available to the program), is associated with each byte for odd parity checking purposes. The USAFETAC IBM 360/44 has a total memory capacity of 262,144 bytes (256K, K = 1024 decimal). Because of the optimization of Model 44 toward scientific usage, the common units of data are the 32-bit binary word and the 16-bit binary half-word, composed of 4 and 2 bytes, respectively. Floating-point arithmetic uses the 32-bit word for short operands and a 64-bit double word for long operands. For more information about internal data representation and word formats refer to IBM Manual GA22-6821, IBM System/360 Principles of Operation. In addition, this information is condensed on the IBM Reference Data card (the "green card"), copies of which are available in the Systems Section of the Data Automation Branch.

Processor storage speed for the Model 44 is 1 microsecond. In each access, 4 bytes (one word or two half-words) are stored or fetched.

1.3. TELEX 5312 Direct Access Storage Devices (DASD). The seven TELEX 5312 DASDs are used in the system as input, output, and storage devices. Because of the physical characteristics of direct access, they are able to process data records in a completely random method.

The TELEX 5312 uses IBM 2316 or equivalent disk packs that have a total of 200 cylinders for system or programmer use. Each cylinder contains a total of 20 tracks with a maximum capacity per track of 7294 bytes. A disk can, therefore, hold a total of 29.17 million bytes of data. The system can access this data at a transfer rate of 312K bytes per second.

The disk devices are used by the system for its libraries, spooling, and work files. The programmer can use them for data and work files. To meet the requirements of both the system and the programmer, the Systems Section has allocated file space on the disk packs and assigned them to the seven disk drives. For a complete description of USAFETAC's disk assignments and organization refer to Section 2.3 in this guide.

1.4. IBM 1052 Console Printer/Keyboard. The console permits communications between the operator and the system. The programmer may communicate with the operator using the Job Control Statements PAUSE and * TEXT or, from the program, by writing a message to the console.

Programmer-generated console messages must be both concise and meaningful. They must be controlled to maintain standardization. Before any program is considered to be operational, the Program Librarian approves and catalogs all console messages used. The Program Librarian maintains a listing of all cataloged console messages.

All messages are of the form: aNNNb text
Where:

```
    a = F if the message is written by a subprogram
        H if the message is written by a main program
  NNN = a 3-digit number assigned by the Systems Section
    b = A if operator action is required
        D if operator decision must be made and action taken
        I if for operator information only
 text = the message text
```

By using this format for all console messages, a program can give more information to the operator than would be possible with a simple text message. As new programs are developed, a quick check of the message listing may prove that a prior message suitable for your use has been cataloged by the Program Librarian.

USAFETAC has only one console printer/keyboard which must be shared by all partitions so console I/O should be kept to a minimum. The device address for all partitions is X,'01F'. The logical name is SYSLOG. These are the system default assignments. The data set reference number for FORTRAN WRITE is 15.

1.5. Magnetic-Tape Units. The USAFETAC System/360 has six 9-track tape drives and two 7-track tape drives. The magnetic-tape units function in the system as both input/output devices and data storage units. They transport the tape and read and write the information as directed by the system.

1.5.1. STC 3470 Magnetic-Tape Unit (9-Track). The six STC 3470 Magnetic-Tape Units write data on 1/2-inch wide magnetic tape in a 9-track format. The standard tape density is 1600 bytes per inch (BPI) with an optional 800 BPI. A 1600-BPI tape written in the standard USAFETAC Data Base format can contain 35 million bytes of data.

These tape units have a data transfer rate of 320,000 bytes per second at a recording density of 1600 BPI. A tape-transport rate of 200 inches per second is standard for both recording densities. The interblock gap of 0.6 inch is also standard.

1.5.2. IBM 2401 Magnetic-Tape Unit (7-Track). The two IBM 2401 Magnetic-Tape Units write data on 1/2-inch wide magnetic tapes in the IBM 729 7-track format. The standard tape density is 800 BPI with 200- and 556-BPI densities optional.

The tape units have a data transfer rate of 60,000 bytes per second at a recording density of 800 BPI. The tape transport rate is 75 inches per second for all three recording densities.

1.5.3. Tape Assignments. There are no standard assignments for the tape drives. The user must temporarily assign them for each job with the Job Control Statement;

```
// ASSGN SYSnnn,X'cuu',x'ss'
```

where SYSnnn is the logical name, cuu is the physical-device address and ss is the data mode.

To eliminate possible logical name conflict and to standardize assignments, only the following programmer logical assignments should be used:

FORTRAN - SYS005-SYS011 for data-set reference numbers 8-14, respectively.

COBOL and ASSEMBLY Programs - SYS013-SYS029 as determined by the FILE-CONTROL and DTF, respectively.

SYSTEMS SUBPROGRAMS - Logical names SYS030-SYS049 are reserved for USAFETAC system's subprograms use.

The physical device addresses for the six 9-track tape drives are 1A0, 1A1, 1A2, 1A3, 1A4, and 1A5. The two 7-track drives are physical device addresses 1E0 and 1E1. To reduce interpartition competition for tape drives as much as possible, the physical addresses should be assigned in ascending order (1A0-1A5 and 1E0-1E1) in background (BG), and descending order (1A5-1A0) and (1E1-1E0) in foreground (F2). In addition, output tapes are assigned, by convention, to the outer drives, while input tapes are assigned to the inner drives. For BG, 1A0 is the outer drive; for F2, 1A5.

For a more complete description of the ASSGN statement, refer to IBM Manual GC24-5036, DOS Systems Control and Service.

1.6. IBM 2540 Card Read/Punch. The IBM 2540 Card Read/Punch is an input/output device for the IBM System/360 Model 44. It has separate read and punch feeds with maximum reading speed of 1000 cards per minute and maximum punching speed of 300 cards per minute. The punch hopper holds approximately 1350 cards; the read hopper and file-feed magazine hold approximately 3100 cards.

USAFETAC has only one card read/punch; however, the phantom device function of the GRASP spooling system makes it appear as two card readers. The standard assignments for the card read/punch are as follows:

Physical Address BG	Physical Address F2	Logical Name	FORTRAN Data Set
00C	02C	SYSIPT (for data) SYSRDR (for JCL)	1
00D	00D	SYSPCH	2 (or PUNCH)

1.7. IBM 1403 Model N1 Printer. The printer is an output device used for listing Job-Control Statements, compiler/assembler/linkage editor output, and program output. It uses fanfold paper, preprinted forms, or adding machine type paper tape. USAFETAC has only one actual printer; however, as with the card reader, GRASP provides for a second "phantom" printer.

A printer's rated speed is based upon the number of single-spaced lines that can be printed per minute. Actual printing speed depends on the character set used and the time required for processing and moving paper. For the Model N1, the maximum printing speed is 1100 lines per minute.

The basic character arrangement for the printer is the standard Extended Binary-Coded-Decimal Interchange Code (EBCDIC). Each of the 132 print positions can contain one of 60 different characters; 26 alphabetic, 10 numeric, and 24 special characters. (The FORTRAN programmer should be aware that the first character of the output record is not printed, but is used instead for printer carriage control. The second character is then printed in print position one. For this reason, FORTRAN I/O allows for an output record of 133 bytes to include the control character).

Horizontal spacing is 10 characters per inch. Standard vertical spacing is six lines per inch with an optional operator-controlled eight lines per inch. Vertical spacing and skipping are controlled by the user program. Standard skipping rate is about 33 inches per second. The carriage is dual speed and permits high-speed skipping at about 75 inches per second on skips of more than eight lines.

The standard system assignments for the printers (actual and phantom) are:

Physical Address BG	Physical Address F2	Logical Name	FORTRAN Data Set
00E	02E	SYSLST	3 (or PRINT)

Chapter 2

SYSTEM ORGANIZATION

2.1. General. The IBM Disk Operating System (DOS) consists of the control programs and one or more processing programs. The processing programs are written by USAFETAC programmers or they are supplied by IBM or other commercial vendors. The system is disk resident, accessed through TELEX 5312 disk drives.

The control program is that part of DOS that prepares and controls the execution of the other programs in the system. It includes the Supervisor (including IOCS), Job Control, and SDI's GRASP (see Section 2.4).

Main storage for the USAFETAC IBM 360/44 contains a total of 256K bytes. The Supervisor occupies the low 16K bytes of main storage. The 160K bytes occupied by the background (BG) partition begin just past the transient area of the Supervisor. Following the BG partition is the foreground-two (F2) partition occupying 62K bytes. The remaining 18K bytes is the foreground-one (F1) partition in which GRASP resides.

2.2. Supervisor. The Supervisor is that part of the control program that handles all input/output operations, interruption conditions, and other functions for all problem programs. The functions performed by the Supervisor are:

a. Storage protection (required for multiprogramming)
b. Interrupt handling
c. Device error recovery
d. Collection of tape error statistics by volume
e. Error volume analysis
f. Error logging and recovery
g. Operator recovery
h. Program retrieval (fetch and load)
i. End-of-Job handling
j. Timer service
k. Checkpoint/Program restart

All functions, except certain interrupt handling (SVC, I/O, and machine check), are available to the problem program by issuing macro instructions. The programmer is not concerned with machine interrupt conditions, since these are handled automatically by the Supervisor.

Part of the Supervisor resides in main storage at all times to act as the system controller. Certain functions of the Supervisor are provided by transient routines that remain in disk storage until needed. When needed, the Supervisor loads them into the transient area (overlaying the previous routine in the area).

Control is temporarily relinquished by the Supervisor to the active program having the highest priority. Priority of programs in the system has been assigned as follows:

a. Supervisor (highest priority)
b. System operator communication routine
c. Foreground-one program (F1 - GRASP)
d. Foreground-two program (F2)
e. Background program (BG) - (lowest priority)

2.3. Job Control. The Job Control program provides job-to-job transition for all batched jobs running within the disk operating system. It is also called into main storage to prepare each job step to be run. (One or more programs can be executed within a single job. Each such execution is called a job step.) It performs its functions between job steps and is not present while a problem program is executing. Job Control is called by:

a. The Initial Program Loader, to process the first job after IPL, and
b. The Supervisor, at normal end of each job step or at abnormal end of job.

Job Control performs various functions on the basis of information provided in

Job Control Statements. These functions are:

a. Prepare programs for execution (see Section 2.3.2)
b. Assign device addresses to symbolic names (see Section 2.3.2)
c. Set up fields in the communications region
d. Edit and store volume and file lable information
e. Provide the interface with a user written output routine for job accounting.

2.3.1. Preparing Programs for Execution. All programs run in the system are loaded from the core image library on the resident disk pack. If a program has been previously cataloged, Job Control constructs a problem program phase directly on SYSRES and directs the system loader to load that program for execution.

2.3.2. Symbolic I/O Assignment. Job Control is responsible for assigning physical I/O units. Programs do not reference I/O devices by their actual physical addresses, but rather by symbolic names. The ability to reference an I/O device by a symbolic name rather than a physical address provides advantages to both programmers and machine operators. The symbolic name of a device is chosen by the programmer from a fixed set of symbolic names. He can write a program that is dependent only on the device type and not on the actual device address. At execution time, the operator or programmer determines the actual physical device to be assigned to a given symbolic name. This is communicated to Job Control by a control statement (ASSGN). Job Control associates the physical device with the symbolic name by which it is referenced.

A fixed set of symbolic names is used to reference I/O devices. No other names can be used. They are:

SYSRDR Card reader, magnetic tape unit, or disk extent used for Job Control Statements.

SYSIPT Card reader, magnetic tape unit, or disk extent used as the input unit for programs.

SYSPCH Card punch, magnetic tape unit, or disk extent used as the main unit for punched output.

SYSLST Printer, magnetic tape unit, or disk extent used as the main unit for printed output.

SYSLOG Console printer-keyboard used for operator messages and to log Job Control Statements.

SYSLNK Disk extent used by compilers and assemblers to generate input to the Linkage Editor.

SYS001-SYS004 Disk extents used for system compiler and assembler work files.

SYS005-SYS011 Used by FORTRAN programs (for data set reference numbers 8-14).

SYS013-SYS029 Used by COBOL and ASSEMBLY programs as determined by the FILE-CONTROL and DTF, respectively.

SYS030-SYS049 Used by USAFETAC systems subprograms only.

The first six of these symbolic names, termed system logical units, are used by the System Control program and System Service programs. Of these six units, user programs may also use SYSIPT and SYSRDR for input, SYSLST and SYSPCH for appropriate output, and SYSLOG for operator communications. Normally, SYSRDR and SYSIPT both refer to the same device. The remaining symbolic names are termed programmer logical units. These may be used by the programmer, within the limits specified, for any device as appropriate.

2.3.3. Job Control Statements (JCS). Job Control Statements, normally entered by the programmers, are used for batched-job programs only. They are usually coded as part of the input job stream and entered through SYSRDR.

Certain rules must be followed when writing Job Control Statements:

a. Name. Two slashes (//) identify the statement as a Job Control Statement. They must be in columns 1 and 2. At least one blank immediately follows the second slash. Exception: The End-of-Job statement contains /& in columns 1 and 2, the End-of-Data statement contains /* in columns 1 and 2, and the Comment statement contains an * in column 1 and blank in column 2.

b. Operation. This describes the type of control statement (the operation to be performed). It can be up to eight characters long. At least one blank follows its last character.

c. Operand. This may be blank or may contain one or more entries separated by commas. The last term must be followed by a blank, unless its last character is in column 71. Any blank within the operand fields, except for fields contained within apostrophes, is considered an end-of-operand indication. No further processing of that card occurs.

All control statements are essentially free form. Information starts in column 1 and cannot extend past column 71. Continuation cards are not recognized by Job Control.

Job Control normally reads from the device identified by the symbolic name SYSRDR. However, Job Control Statements can also be entered through SYSLOG. For a complete description of the various Job Control commands, including formats, refer to IBM Manual, GC24-5036, DOS Systems Control and Services.

2.4. GRASP. An additional software package developed by Software Design Inc. (SDI), GRASP, is installed in the 360/44. It operates in the foreground-one (F1) partition above the Job Control program and the Supervisor. During input spooling, GRASP reads each JCS card first, acting on those cards containing GRASP statements and passing the others to the Job Control program. During output, GRASP automatically spools printer output to decrease CPU dependence on the significantly slower printer. GRASP offers three very helpful facilities to the programmer: EPROC, GRASPCTL, and LINEUP.

2.4.1. EPROC. The extended procedures feature of GRASP (EPROC) allows the programmer to copy card images from the source statement library into the job stream. These cards may be data cards, standard JCS cards, or program source statement cards. This feature presents several benefits: 1) It reduces the number of cards keypunched by the programmer; 2) It eliminates errors such as card read errors, inadvertent card shuffling, card failure due to frequency of use, etc. Any contiguous collection of cards having a high frequency of use (except program source books) should be considered a potential candidate for being cataloged as a USAFETAC PROCEDURE and given to the Program Librarian for approval and cataloging. Once the JCL for a routinely executed production program has been cataloged as a procedure, the programmer need submit only the // JOB card, one or two cards to invoke the procedure, any required data cards, and the End-of-Job card. For example, suppose the following book had been cataloged into the procedure library (sublibrary G.):

```
   BKEND G.ADSPROGM
// OPTION NOLOG                              0000
// PAUSE *** MOUNT TAPE FOR ADSPROGM         0001
// ASSGN SYS005,X'1A5'                       0002
// MTC REW,SYS005                            0003
// MTC WTM,SYS005,2                          0004
// MTC REW,SYS005                            0005
// EXEC ADSPRGM1                             0006
%/ DATA                                      0007
%/ EPROC G.SLASHAST                          0008
// MTC REW,SYS005                            0009
// EXEC ADSPRGM2                             0010
// MTC RUN,SYS005                            0011
   BKEND
```

The following card deck would be all that is required to execute this procedure:

```
// JOB ADSPROGM,(050410-MLF-10-1C),A=F2
// EPROC ADSPROGM
1234 AB DATA CARD
// EPROC ,
/&
```

The following actions are taken by GRASP for the above example:

The // JOB card is passed as is to Job Control.
The // EPROC ADSPROGM card is replaced by the cards in the cataloged EPROC book down to the %/ DATA card and passed to Job Control.

The user's data card 1234 AB DATA CARD is passed as is to Job Control. Note: A %/ DATA card indicates to GRASP that any number of data cards on SYSIPT down to the next null EPROC card (// EPROC ,) are to be inserted.

The remaining cards in the cataloged EPROC book are passed to Job Control. The %/ EPROC G. SLASHAST card is a nested EPROC and causes GRASP to retrieve from the SLASHAST book the /* card (which is the only card in the book).

Finally, the user's /& card is passed as is to Job Control.

This example showed replacement with the entire cataloged book. However, the programmer may reference any portion of any cataloged book simply by specifying the book's sublibrary (if omitted, the default is G.), book name, and beginning and ending sequence numbers. For example:

```
// JOB NEWONE etc.
// EPROC G.ADSPROGM,0000,0005
// EXEC ADSPRGM3
// MTC RUN,SYS005
```

Use of this job deck could save recutting JCL cards that are the same as a string already cataloged. This procedure is very helpful also in creating a new program from pieces of already-cataloged source programs.

2.4.2. GRASPCTL. Another useful GRASP feature, GRASPCTL, allows the programmer to instruct the spooling program to produce multiple copies of a program's output. This is accomplished by including the cards:

```
// EXEC GRASPCTL
   SET SYSLST,COPIES=mmm
```

where SYSLST is optional and is the logical assignment and mmm is the number of copies desired. If the spooling buffer cannot hold the entire multi-copied output, only one copy is produced and a message is issued to that effect. The multi-copy option is terminated at End-of-Job. (Note: Be sure to indicate on the job submission form that GRASPCTL has been set and the number of copies requested.)

2.4.3. LINEUP. When the size of the output file for which multiple copies are desired would exceed 50 printed pages or the programmer requires a special form, the programmer should use another GRASP feature -- LINEUP. This option is used to instruct the operator to load multi-ply paper or preprinted forms in the printer. Because of the nature of spooling, use of the // PAUSE card in the job stream would be incorrect. LINEUP on the other hand causes the change-paper message to be sent to the operator at the time the spooling program, not the user-executed program, is ready to deliver that job's output to the printer. The following is an example of the use of LINEUP:

```
// JOB ETC.
// ASSGN etc.
// EXEC LINEUP
 * PLEASE LOAD TWO PLY PAPER IN THE PRINTER
// EXEC PROGRAM
```

```
// EXEC LINEUP
 * PLEASE RETURN PRINTER TO SINGLE COPY
/&
```

(NOTE: The programmer may use GRASPCTL and LINEUP together when necessary to obtain a balance between legible multi-ply copies and output file size.)

2.5. Disk Organization. The IBM Disk Operating System provides positive identification and protection of all DASD files by recording labels on each volume (disk pack). These labels insure that the correct volume is used for input and that no current information is destroyed on output.

The standard volume label identifies the entire volume. The Systems Section writes this label with the Initialize Disk Utility program at the time the disk pack is prepared for use. Every volume used in the disk operating system environment must have a standard volume label. The information in the standard volume label is checked, but never altered, by IOCS during file processing.

A standard file label identifies a particular logical file, gives its location(s) on the pack, and contains information to prevent premature destruction of current data (if the file is to be permanent).

Because each file label contains file limits, the group of labels is essentially a directory of all data records on the volume. All standard file labels are grouped together and stored in the Volume Table of Contents (VTOC) on the disk pack. The Systems Section maintains current listings of all VTOC's. Programmers requiring either temporary or permanent disk storage must contact the Systems Section for space allocation.

All DASD label processing is performed by the transient label processing routines of the Supervisor. These routines use the information stored on the label information cylinder on the resident pack. This information is supplied by the DLBL and EXTENT Job Control cards. A DLBL card must be supplied for each logical file, and an EXTENT card must be supplied for each extent (area) in which the file is located.

When a job step requires access to a file on one of the disk volumes, the DLBL and EXTENT cards in the job stream are read into the appropriate area on the label information cylinder. During execution, when the program attempts to open the desired file, label checking occurs.

Files thus identified exist throughout that and each succeeding job step until the next DLBL or until the End-of-Job. For a complete description of DLBL and EXTENT parameters, refer to the IBM Manuals DOS Systems Controls and Services and DOS DASD Labels. (NOTE: Because of the nature of DOS data management, do not change any parameters of a DLBL or EXTENT statement without prior coordination with the Systems Section.)

Three of the seven disk drives are permanently mounted and assigned in the system. The programmer is not required to enter DLBL, EXTENT, or ASSGN statements for the job to access the system files. However, permanent data files may require one or more Job Control Statements. Any such requirements will be documented in the programs that require such access. The three permanent volumes are:

SYSRES pack on 280 in both partitions contains the system libraries, EPROC library and the error logging files. The background core image library (program start address X'4000') is stored here.

SYSLIB pack on 281 in both partitions contains the update source statement library and some permanent data files. Also, the foreground core image library (program start address X'2C078') is stored here.

SYSFIL pack on 285 in both partitions contains permanent data files. These are mostly standard data base type files for which many different programs require routine access. Also, the GRASP spooling files and the accounting files are located here.

The remaining four disk drives are used for packs that are not permenently assigned in the system. The programmer must include the ASSGN, DLBL, and EXTENT cards to access any file on these packs (except as noted below).

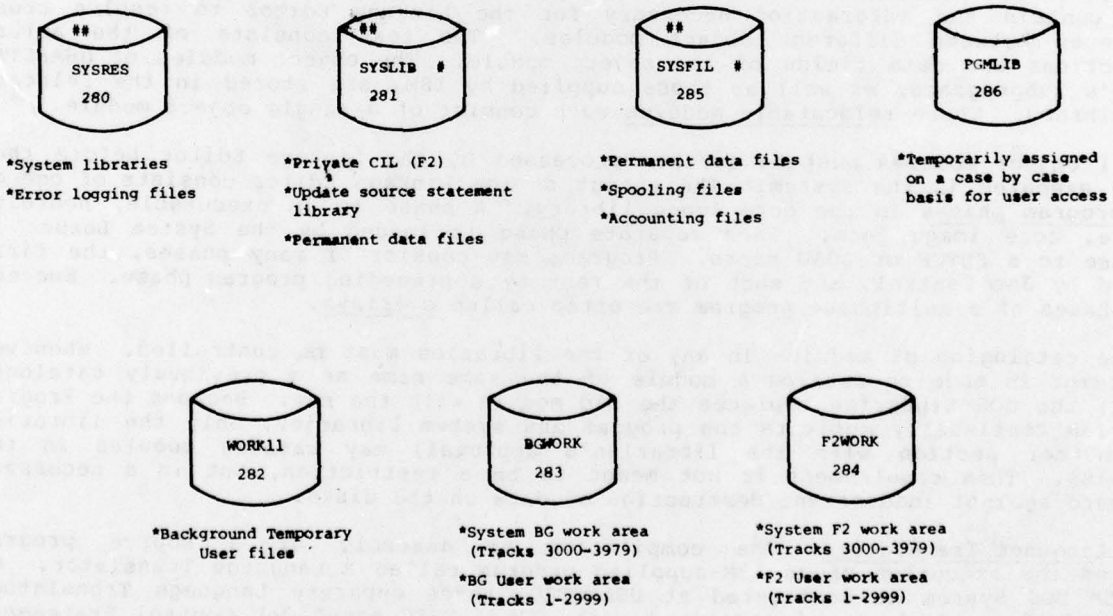
WORK11 pack on 282 in background only. Tracks 1-3979 may be used by the programmer for temporary work files. These files (and those on the next two packs) will only exist for the current job.

BGWORK pack on 283 in background only.

F2WORK pack on 284 in foreground only. The programmer has access to tracks 1-2999 during any job step. The system uses tracks 3000-3979 as work files for the language translators and the Linkage Editor. The programmer may make DLBL and EXTENT changes that use tracks 1-3979 in any job that doesn't compile, assemble or link or in any job step after program compilation and linkage have finished.

An extra pack on 286 in both partitions is used on a case by case basis for user access. (See the Systems Section in the Data Automation Branch for pack assignments.)

Figure 2-1 summarizes the preceeding information for handy reference.



\# See the Systems Section in the Data Automation Branch to obtain space for permanent data files.

\#\# These are permanently assigned and their use is transparent to the programmer.

Figure 2-1. USAFETAC Disk Pack Assignments.

Chapter 3

SYSTEM/360 PROGRAMMING

3.1. General. In addition to the control facilities of the Supervisor and Job Control programs, the Disk Operating System also provides certain programs required for processing user's software. Language translators are supplied to convert program source code into relocatable object code. Service programs are used to edit problem programs into executable form (Linkage Editor) and maintain and service the system resident libraries (Librarian). Another service function, Sort/Merge, enables users to sort multiple files of randomly ordered records or to merge multiple files of sequenced records into one sequential file. Utilities are provided to copy data files from one storage medium to another or to initialize disk volumes and files for programmer use. Figure 3-1 presents a hierarchical view of the software facilities that comprise the System/360 DOS at USAFETAC.

The term program could be confused with several things. The programmer codes sets of source statements that may be a complete program or part of a program. These source statements are then compiled or assembled into a relocatable machine-language program which, in turn, must be linked into an executable program, and may be combined with other programs. Consequently, it is convenient to refer to each state of program development by a particular name.

A set of source statements that is processed by a language translator (Assembler, COBOL, FORTRAN, etc.) is referred to as a source module. A source module consists of definitions for one or more control sections. When the source module is translated, the output (object module) consists of one or more defined control sections. Each control section is a block of code assigned to contiguous main-storage locations. The input for building a phase (a section of a program loaded as a single overlay) must consist of one or more complete control sections.

The output of a language translator is referred to as an object module. It consists of the dictionaries and text of one or more control sections. The dictionaries contain the information necessary for the Linkage Editor to resolve cross references between different object modules. The text consists of the actual instructions and data fields of the object module. The object modules of USAFETAC System's subprograms, as well as those supplied by IBM, are stored in the relocatable library. These relocatable modules each consist of a single object module.

All object modules must be further processed by the Linkage Editor before they can be executed in the system. The output of the Linkage Editor consists of one or more program phases in the core image library. A phase is in executable, nonrelocatable, core image form. Each separate phase is loaded by the System Loader in response to a FETCH or LOAD macro. Programs may consist of many phases, the first fetched by Job Control, and each of the rest by a preceding program phase. Successive phases of a multiphase program are often called overlays.

The cataloging of modules in any of the libraries must be controlled. Whenever an attempt is made to catalog a module of the same name as a previously cataloged module, the DOS Librarian replaces the old module with the new. Because the Program Librarian continually monitors the program and system libraries, only the librarian (or another section with the librarian's approval) may catalog modules in the libraries. This requirement is not meant to be a restriction, but is a necessary safeguard against inadvertant destruction of data on the disks.

3.2. Language Translators. The compilation or assembly of a source program involves the execution of an IBM-supplied program called a Language Translator. On the IBM DOS System as configured at USAFETAC, three separate Language Translators are available. Each one is executed with a "// EXEC name" Job Control Statement. The names used to call one of the three possible translators are:

ASSEMBLY	Calls the IBM Assembler Program
FCOBOL	Calls the American National Standard COBOL Compiler
FFORTRAN	Calls the FORTRAN IV Compiler

The following sections give more information about the use of each language translator.

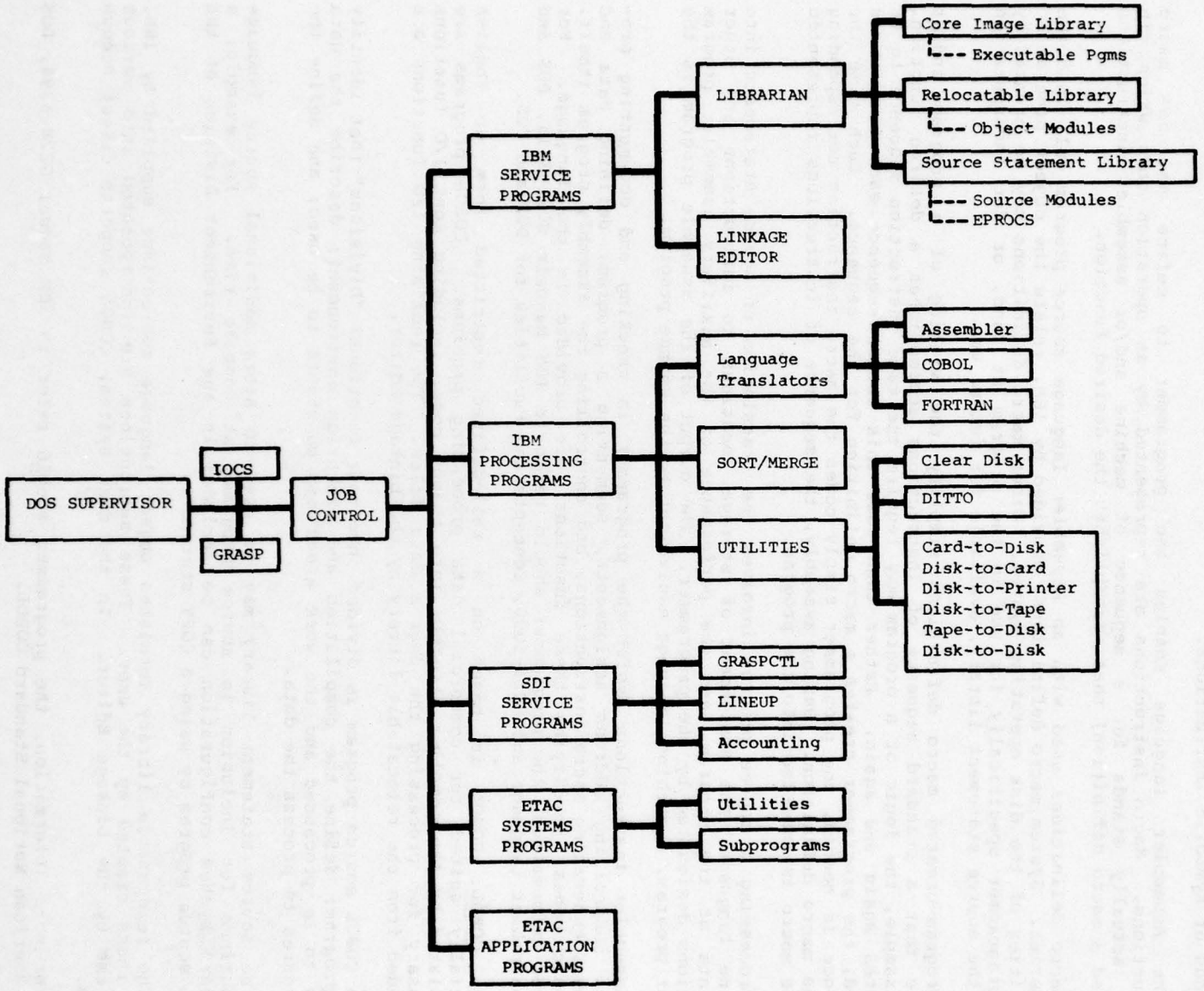


Figure 3-1. USAFETAC System/360 DOS Facilities.

3.2.1. Assembler. The Assembler language is a symbolic, machine-oriented language that is applicable to any problem. The problem program coding is done with symbolic instructions that are translated into machine instructions. Program locations can be addressed through symbolic names. Data constants can be defined in several different ways, either as explicit constants or as literals coded directly into the operands of symbolic instructions.

The Assembler language enables the programmer to define and use macro instructions. Macro instructions are represented by an operation code which, in turn, actually stands for a sequence of machine and/or assembler instructions (called a macro definition) that accomplishes the desired function.

Macro definitions used with an Assembler language source program fall into two categories. System macro definitions, provided by IBM, relate the object program to facilities of the disk operating system. Other macro definitions may be created by the programmer specifically for use in the program at hand, or for incorporation into the source statement library, available for future use.

Program-created macro definitions simplify the writing of a program and/or ensure that a standard sequence of instructions accomplishes a desired function. For example, the logic of a program may require the same instruction sequence to be executed again and again. Rather than code this entire sequence each time it is needed, the programmer creates a macro definition for the sequence. Each time the sequence is needed, the programmer simply codes the macro instruction corresponding to the macro definition. During assembly, the sequence of instructions represented by the macro is inserted into the program.

Processing a source program involves the translation of source statements into machine language, the assignment of storage locations to instructions and other elements of the program, and the performance of the auxiliary assembler program functions designated by the programmer. The output of the assemble program is the object program, a machine language equivalent to the source program.

Assembler instructions assist the programmer in checking and documenting programs, controlling address assignments, segmenting a program, defining data and symbols, generating macro instructions, and controlling the assembly program itself. Mnemonic codes, specifying these functions, are provided in the language. For further information, the programmer should refer to IBM Manuals GC24-3414, DOS and TOS Assembler Language and GC24-5030, Concepts and Facilities for DOS and TOS.

3.2.2. COBOL. COBOL is based on a well-defined restricted form of English especially suited for commercial data processing problems. COBOL programs are translated by the compiler directly into object code, including most I/O functions necessary for processing the user's data files. The remaining I/O functions are included from the relocatable library by the Linkage Editor.

A COBOL source program is divided into four functional "Divisions" that identify the program; define the compilation and execution environment; describe the data files to be processed and the work areas and constants to be used; and define the procedures to process the data.

The source statement library may be used to store additional source language subroutines for inclusion in source programs at compile time. For example, a standard system configuration can be included in the Environment Division of the COBOL source program by using a COPY statement.

The relocatable library furnishes object language subroutines supplied by IBM, plus those created by the user. These subroutines are incorporated into various programs by the Linkage Editor. In the disk system, COBOL supports direct access files.

For more information, the programmer should refer to IBM Manual GC28-6394, DOS Full American National Standard COBOL.

3.2.3. FORTRAN. The Disk Operating System provides an IBM FORTRAN compiler in addition to its basic FORTRAN facility. The FORTRAN compiler requires a minimum partition of 40K for compilation; thus, it may be run in either partition. The FORTRAN language is compatible with and encompasses the American National Standard FORTRAN. All basic FORTRAN-IV facilities apply to FORTRAN. In addition, several extensions to standard FORTRAN are available. For a complete list, refer to the IBM FORTRAN IV Language Manual.

Because of the scientific nature of most applications programming at USAFETAC, the Program Librarian has on tape the entire Scientific Subroutine Package (SSP). Some of these subroutines have already been cataloged in the system relocatable library. If a programmer determines that one of the SSP modules will be required, the programmer may ask the librarian to catalog that routine in the system. Using this method insures that only those modules actually required at USAFETAC are cataloged.

As an aid for FORTRAN programmers, the Systems Section has assembled a "FORTRAN Programmer's Guide," which is available on a one-per-section basis. This guide is comprised of the four IBM manuals that are required to write FORTRAN programs at USAFETAC:

a. Concepts and Facilities for DOS and TOS

b. DOS Systems Control and Service

c. FORTRAN IV Language

d. FORTRAN IV Programmer's Guide

The Systems Section suggests that each newly assigned programmer become thoroughly familiar with each of these manuals. Whenever questions arise or 'bugs' appear in a program, the programmer should be able to solve most of the problems by referencing these manuals.

Figure 3-1 illustrates the use of specific devices within the FORTRAN facility. The figure equates the data set reference numbers with the device types, logical and file names, and physical addresses. The logical name is used on the EXTENT and ASSGN JCS cards while the file name is used on the DLBL and TLBL cards.

3.3. Linkage Editor. All programs executed in the Disk Operating System environment must be edited by the Linkage Editor. The Linkage Editor reads the relocatable output of the language translators and edits it into executable, nonrelocatable programs in either the system core image library (background) or a private core image library (foreground). The Linkage Editor performs on one program at a time; that is, it cannot linkage edit a series of programs concurrently. Once a program is edited, it can be executed immediately, or it can be cataloged as a permanent entry in a core image library (by the Program Librarian only). When a program has been cataloged in a core image library, the Linkage Editor is no longer required for that program. The program is run as a distinct job step and is loaded directly from a core image library by the System Loader.

The extent of the editing function performed depends on the structure of the input program. The simplest case is that of a single-module program. The Linkage Editor has only to edit the program, creating a single phase entry in the core image format.

In more complex situations, the operation may involve linking together and relocating multiple-control sections from separate assemblies to produce a number of separate phases. The Linkage Editor resolves all linkages (symbolic reference) between segments of the program and relocates the phases to load at specified main-storage locations.

To facilitate writing and testing large programs, assembled program modules cataloged in the relocatable library can be combined with other modules from SYSIPT (card, tape, or disk).

The Linkage Editor is run as a distinct job step. Because of this fact, it is meaningful to classify it as one of the System Service Programs (See Figure 3-1). The Linkage Editor function is performed as a job step in three kinds of operations.

3.3.1. Catalog Programs in Core Image Library. The Linkage Editor function is performed immediately preceding the operation that catalogs programs into a core image library. By specifying the CATAL option, the Linkage Editor not only edits the programs, but also catalogs them permanently in the core image library.

The input for the LNKEDT function could include modules from a relocatable library instead of, or in addition to, those modules from the card reader, tape unit, or disk extent assigned to SYSIPT. This is accomplished by including the name of the module to be included in an INCLUDE statement.

3.3.2. Load-and-Execute. Specifying OPTION LINK causes Job Control to open SYSLNK and allows Job Control to place the object module(s) and Linkage Editor control statements on SYSLNK. Just as with the catalog operation, the input can consist of object modules from a relocatable library instead of, or in addition to, those input through SYSIPT. This is accomplished by including the name of the module to be included in the operand of an INCLUDE statement. After the object modules have been edited and placed in a core image library, the program is executed. The blank operand in the EXEC control statement indicates that the program just linkage edited and temporarily stored in a core image library is to be executed.

3.3.3. Assemble- or Compile-and-Execute. Source modules can be assembled or compiled, linkage edited and then executed in a single sequence of job steps. In order to do this, the language translator is directed to output the object module directly to SYSLNK. This is done by using the LINK in the OPTION control statement. Upon completion of this output operation, the Linkage Editor function is performed. The program is linkage edited and temporarily stored in a core image library.

In addition to the program cards previously listed, object modules used as input for the Linkage Editor include Linkage Editor control statements. There are four kinds of these control statements (phase, include, entry, and action). For specific use of these control statements and examples of Linkage Editor 'job decks' consult IBM Manual GC24-5036, DOS Systems Control and Service.

3.4. Program Library. The USAFETAC Program Librarian maintains all production programs and the source decks and listings for all production programs, subprograms, utilities and procedures at USAFETAC. Program source decks are normally maintained in the source statement library on the SYSPGM disk pack. Programs to be updated are moved to another source library by the Program Librarian and are then available for modification by programmers and analysts. The source code for every production version of all USAFETAC programs and subprograms is also archived on tape.

The Program Librarian maintains directory listings of all system and program libraries. These listings are updated daily for the update source statement library and the archived source statement library. The core image and relocatable library directory listings are updated at least weekly.

The Librarian also produces listings of all the documentation sections of all USAFETAC software. The documentation for each program is distributed to the user at the time the program is cataloged. In addition, the Librarian distributes complete documentation listings for all subprograms and utilities on an as needed basis.

To help eliminate duplication of programming effort, the Program Librarian also generates a cross-reference listing of all USAFETAC software. The analyst or programmer uses this 'Keyword Library Listing' to locate any software that might have been developed previously to solve the same or similar program problem. The 'keyword' programs produce two basic listings: 1) a listing of all USAFETAC software by keyword and 2) a listing, alphabetically by name, of all USAFETAC programs, subprograms and utilities. Both outputs list the name, language, computer, OPR, status and an 80-character description for each given module. After finding a possible module, the programmer can refer to the documentation listings for further information. If further research is required, the programmer can also visit the Program Librarian to look at the actual program listing. Following these steps can save much of the programmer's time by revealing the existence of software that will solve the program problem.

APPENDIX

LISTING OF APPLICABLE IBM SYSTEM REFERENCE MANUALS

1. GA21-9033
   IBM System/360 Component Description and Operating Procedures
   IBM 2540 Card Read Punch

2. GA22-6866
   IBM System/360 Component Descriptions
   2400-Series Magnetic Tape Units
   2803/2804 Tape Controls and
   2816 Switching Unit, Model 1

3. GA22-6875
   IBM System/360 Model 44 Functional Characteristics

4. GA24-3073
   IBM 1403 Printer Component Description

5. GA26-3599
   IBM System/360 Component Descriptions
   *2314 Direct Access Storage Facility and
   2844 Auxiliary Storage Control

6. GC24-5022
   DOS Operating Guide

7. GC24-5036
   DOS Systems Control and Service

8. GC24-5074
   DOS Messages

9. GC24-5030
   Concepts and Facilities for DOS and TOS

10. GC24-3414
    DOS and TOS Assembler Language

11. GC28-6394
    DOS Full American National Standard COBOL

12. GC28-6515
    IBM System/360 and System/370 FORTRAN IV Language

13. GC24-5073
    DOS System Programmer's Guide

*The IBM 2314 direct access storage facility is functionally compatible with the TELEX 5312, and the programmer may use this manual for reference.

LIST OF ABBREVIATIONS AND ACRONYMS

ASSGN	control statement to assign a device to a job
BG	background
BPI	bytes per inch
COBOL	common business oriented language
CATAL	control option to catalog a program in the program library.
COPY	add a source book into the Assembly stream
CPU	central processing unit
DASD	Direct Access Storage Devices
DLBL	Disk Label
DOS	Disk Operating System
DTF	Define The File macro (Assembler)
EBCDIC	Extended Binary-Coded-Decimal Interchange Code
EPROC	extended procedure of GRASP
EXEC	execute a program (control statement)
EXTENT	control statement to define limits of a disk file
FORTRAN	scientific oriented computer language (formula translation)
FORTRAN WRITE	'write' is a FORTRAN statement to write a record
FETCH	macro to load program but not execute
GRASP	software package leased by SDI
GRASPCTL	multiple copy feature of GRASP
IBM	International Business Machines
INCLUDE	FORTRAN statement to include procedures in programming, also LNKEDT statement to include a relocatable module
I/O	Input/Output
IOCS	Input/Output Control System
IPL	Initial Program Loader
JCL	Job Control Language
JCS	Job Control Statement
LINEUP	multiple copy feature of GRASP involving more than 50 pages
LINK	option to link program to allow execution
LNKEDT	Linkage Editor (IBM service program)
LOAD	the process of moving code into main memory and starting execution
macro	assembler coding
OPR	office of primary responsibility
OPTION LINK	control statement to allow linking by LNKEDT
OPTION PUNCH	option to allow punching of cards
PAUSE	Job Control Statement
PRINT	FORTRAN statement to write a record to the printer
PUNCH	FORTRAN statement to write a record to the card punch
SDI	Software Design Inc
SSP	Scientific Subroutine Package
STC	Systems Technology Corporation
SVC	Assembler Language Supervisor Call
SYSFIL	system disk used for GRASP files (volume serial number of...)
SYSIPT	card reader logical unit for input of data
SYSLIB	system disk
SYSLNK	disk extent
SYSLOG	console printer-keyboard
SYSLST	printer
SYSPCH	card punch
SYSPGM	program disk assigned on temporary basis for use by Systems Manager only
SYSRDR	card reader logical unit for input of JCS
SYSRES	system disk
TELEX	Telex Corp
TEXT	Job Control Statement
TLBL	Tape Label Job Control Statement
TOS	Tape Operating System
USAFETAC	United States Air Force Environmental Technical Applications Center
VTOC	Volume Table of Contents

LIST OF USAFETAC TECHNICAL NOTES

Number	Title	Date
76-2	Some Aspects of Estimating the Probability of Cloud-Free Lines-of-Sight in Dynamic Situations	Mar 76
76-3	Model Output Statistics Forecast Guidance (AD-A037148)	Sep 76
77-1	Listings of Seminars Available from Hq AWS, AWS Wings, and AFGWC	Mar 77
77-2	USAFETAC Data Base Handbook	Dec 77
77-3	Soil Moisture Agrometeorological Services (AD-A047760)	Jun 77
77-4	The Impact of Winds-Aloft Errors on Air-to-Ground Ballistic Ordnance Deliveries	Jun 77
77-5	Computation of Solar Declination, the Solar Azimuth Angle, and the Equation of Time	Sep 77
78-001	Technical Seminars Available from AWS and AWS Wings (Supersedes TN-77-1)	May 78
78-002	A Technique for Estimating Clock Two-Hourly Precipitation Rate Distributions (AD-A059874)	May 78
79/001	USAFETAC Analyst/Programmer Handbook, IBM 360 DOS User's Guide	Mar 79